VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Vyhledávání na podobnost v datech z filmové sociální sítě

Similarity Search in Data from Movie Social Network

2010

Bc. Pavel Lednický

Souhlasím se zveřejněním této diplomové práce dle požadavků čl. 26, odst. 9 *Studijního a zkušebního řádu pro studium v magisterských programech VŠB-TU Ostrava*.

V Ostravě 7. května 2010 .....

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 7. května 2010 .....

Rád bych na tomto místě poděkoval Ing. Radimu Bačovi, Ph.D. za odbornou pomoc, věcné připomínky, rady a vedení při tvorbě této diplomové práce.

Abstrakt

Diplomová práce je zaměřena na analýzu a implementaci programů, které mají za úkol stahovat data z internetu a následně stažená data vyhodnotit. První program slouží jako webový robot, který stahuje pro nás důležitá data z internetových stránek Česko-Slovenské filmové databáze. Druhý program slouží k zpracování stažených dat a zobrazení výsledků. Programy jsou napsány v jazyce C#, pro psaní bylo využito vývojové prostředí Microsoft Visual Studio 2008 a pro spravování databáze byl využit MS SQL Server.

Klíčová slova: sociální síť, Česko-Slovenská filmová databáze, webový robot, podobnost, metrika, hodnocení, MS SQL Server, dotaz

Abstract

Thesis is focused on analysis and implementation of programs which are designed to download data from the Internet and then downloaded the data to evaluate. The first program serves as a web robot, which is important to us downloading data from websites Czech-Slovak film database. The second program is to the processing of downloaded data and display results. Programs are written in C# for writing was used development environment Microsoft Visual Studio 2008 and for managing database were used Microsoft SQL Server.

Keywords: social network, Czech-Slovak film database, web robot, similarity, metrics, score, MS SQL Server, query

Seznam použitých zkratek a symbolů

CLR	–	Common Language Runtime
CLS	–	Common Language System
CTS	–	Common Type System
ČSFD	–	Česko-Slovenská filmová databáze
FCL	–	Framework Class Library
FDb	–	Filmová databáze
IMDb	–	Internet Movie Database
LINQ	–	Language Integrated Query
MS SQL	–	Microsoft SQL
SEO	–	Search Engine Optimization
SQL	–	Structured Query Language

Obsah

Seznam tabulek

Seznam obrázků

1 Úvod

Tato diplomová práce je zaměřena na analýzu dat uživatelů filmové sociální sítě a sestavení metriky, jenž by umožnila porovnat podobnost dvou uživatelů na základě jejich hodnocení filmů. Pro jednotlivé uživatele může být totiž zajímavé nalezení jiných uživatelů s podobným vkusem.

V dnešní době již existuje několik sociálních sítí, které se zaměřují na filmovou tématiku. Za dobu své existence už stačily získat mnoho uživatelů, kteří na jejích stránkách uvádějí, jak se jim dané filmy líbí. Malý úvod k sociálním sítím obsahuje druhá kapitola. V této kapitole jsou také zmíněny některé filmové sociální sítě i s přehledem nabízených funkcí a možností.

Pro získání všech potřebných dat z filmové sociální sítě je potřeba buď mít přímo přístup k databázi dané sítě nebo data vyčíst přímo ze stránek, kde jsou prezentována. K získání dat ze stránek je vhodné použít vyhledávacího webového robota, který důležité informace uloží do naší databáze. Více o webových robotech se dozvíte ve třetí kapitole. Před samotným použitím webového robota je nutné si stanovit, která data jsou potřebná a mají se stahovat.

Po sezbírání všech důležitých dat přichází na řadu jejich vyhodnocení. Porovnání hodnocení jednotlivých uživatelů však může být obtížný úkol, jelikož uživatelé obvykle nehodnotí stejné filmy. Cílem této práce je pokusit se sestavit metriku, jenž by umožnila porovnat podobnost dvou uživatelů na základě jejich hodnocení filmů. Kapitola čtyři obsahuje obecný úvod do problematiky metrik a podobnostního vyhledávání.

Pátá kapitola se již zabývá samotnou analýzou dat a programovým řešením cíle diplomové práce. Řešení je rozděleno do bloku pro sběr a uložení dat z filmové sociální sítě (vyhledávací webový robot) a do bloku pro vyhodnocování stažených dat.

V šesté kapitole jsou uvedena a graficky znázorněna naměřená data pro demonstraci průběhu provádění kódu programů. První část je opět věnována práci webového robota pro sběr dat a druhá část zobrazuje úskalí a průběh metod použitých pro porovnávání uživatelů.

V závěru pak jsou shrnuty všechny důležité informace o tomto projektu.

2 Sociální sítě

Systémy sociálních sítí [1] nejsou v podstatě ničím jiným než kombinací specializované webhostingové služby a specializovaného vyhledávače. Pomocí těchto serverů mohou mezi sebou jednotliví uživatelé například komunikovat, sdílet fotky a videa, plánovat akce a srazy, hrát hry, seznamovat se atd. Jednotlivé funkce se samozřejmě u každé sítě liší, tyto základní funkce mají téměř všechny známější sociální sítě. Sociální síť tvoří propojené skupiny lidí, kteří se navzájem ovlivňují. Tvoří se na základě zájmů, rodinných vazeb nebo z jiných důvodů.

Sociální sítě jsou velkým fenoménem dnešní doby a na internetu jich můžeme najít spousty. Téměř tři čtvrtiny lidí mají svůj profil na některé ze sociálních sítí, ať už se jedná o Facebook, Twitter, MySpace nebo Lidé.cz. Čím se od sebe jednotlivé sociální sítě liší? Jsou to služby poskytované uživatelům a jejich zaměření, například na sdílení dat, hledání práce, filmy, hudbu, seznamování atd. Uživatelé se díky jejich pomoci mohou snáze dostat k datům, které je zajímají. Mohou si snadno najít a seznamit se s dalšími osobami se společnými zájmy a bavit se o společných problémech.

Jedna z nejznámějších sociálních sítí v dnešní době je Facebook. Tato síť slouží ke komunikaci mezi přáteli, sdílení videí, obrázků, hudby nebo osobních fotografií. Dalšími obdobnými servery jsou MySpace a česká sociální síť Lidé.cz. Trochu odlišná je sociální síť Twitter, kerá užívá ke komunikaci pouze krátké zprávy a je rozšířena spíše mezi internetovými a počítačovými nadšenci. Další zajímavou sociální sítí je LinkedIn sloužící a usnadňující hledání zaměstnání. Přehled mnoha sociálních sítí můžete najít na této stránce [2].

2.1 Filmové sociální sítě

Jsou to sociální sítě zaměřené svým obsahem a funkcemi na poskytování informací o filmech a dění okolo nich. Patrně nejznámější světovou filmovou sítí je IMDb. Z českých filmových sociálních sítí jsou patrně nejznámější ČSFD a FDb.

2.1.1 IMDb(The Internet Movie Database)

Tato americká sociální síť [3] se zabývá širokým spektrem dění kolem filmů. Nabízí mnoho informací o dění na filmové scéně, jako například informace o filmech, televizních pořadech a seriálech, zprávy okolo filmů, televize, celebrit, nabízí také možnost nahrání zajímavých videí a shlédnutí traileru, také nabízí službu diskuze, kde se mohou lidé vyjádřit k určitým tématům, a mnoho dalších funkcí.

2.1.2 ČSFD(Česko-Slovenská filmová databáze)

Tyto stránky se snaží být českým ekvivalentem americké IMDb. Stránky ČSFD [4] jsou majetkem firmy POMO Media Group s.r.o.a byly založeny v roce 2001. Databáze ČSFD obsahuje 245886 filmů, 39646 herců, 15294 režisérů a 218752 registrovaných uživatelů.
Na úvodní stránce je zobrazeno menu a nejzajímavější aktuální informace nabízené těmito

stránkami. ČSFD nabízí uživatelům možnost výpisu filmů a seriálů v programech jednotlivých televizí. Dále nabízí přehled hraných filmů v českých a slovenských kinech. Uživatel těchto stránek může také nakoupit nebo se jen podívat na filmy právě vycházející na DVD. Při výběru si může uživatel vybrat z těchto možností - premiérová DVD, levné DVD a premiérové Blu-ray disky. Stránky také nabízí stručný přehled zpráv s filmovou tématikou, které se udály. Uživatelé si mají možnost také vytvářet své filmotéky a prohlížet si filmotéky ostatních uživatelů. Stránky dále poskytují funkci bazaru, kde může uživatel uvést film, který by rád sehnal nebo kterého by se rád zbavil. Uživatel si také může procházet seznam dalších uživatelů a prohlížet si jejich profily. Uživatelé zde také mohou psát své názory do diskuze. ČSFD také nabízí žebříček nejlepších a nejhorších filmů a seriálů podle hodnocení uživatelů stránek. Další funkcí stránek je výpis informací uložených v databázi podle uživatelem zadaných kriterií.

Stránky s profily uživatelů si může každý uživatel vyplnit podle svého. Tyto stránky pak mohou obsahovat základní údaje uživatele a to přezdívku, jméno, bydliště, kraj a krátké informace o sobě. Dále si uživatel může na stránku přidat text o své osobě, svých zájmech a vše, co uzná za vhodné. Profilová stránka také obsahuje další odkazy na výpis, například, které filmy uživatel hodnotil, na komentáře k filmům a další informace týkající se filmů přidaných tímto uživatelem. Dále stránka obsahuje odkazy na uživatelovy oblíbené filmy, herce, herečky, režiséry, seriály a spřízněné duše.

Spřízněné duše jsou seznamy dalších uživatelů, kteří mají největší procentuální podobnost v hodnocení filmů nebo seriálů. Zobrazená procenta znamenají podobnost hvězdičkového hodnocení daného uživatele s hodnocením uživatele právě sledovaného profilu. Tedy filmy, u kterých se hvězdičkovým hodnocením shodují, tuto procentuální podobnost zvyšují, a naopak filmy, u kterých se hodnocením rozcházejí, ji snižují. U každé spřízněné duše jsou také uvedeny žánry, ve kterých se shodují oba uživatelé nejvíce. Do této statistiky se započítávají pouze uživatelé s 500 a více hodnoceními a minimálně jedním přihlášením do měsíce. Aktualizace statistiky probíhá každý den, kdy se uživatel přihlásí do ČSFD. Ve spřízněných duších jsou také ještě uvedeny tři porovnání podle shody s oblíbenými režiséry, herci a herečkami daného uživatele. Uspořádání uživatelů se stejným počtem shodných oblíbených tvůrců určuje to, jestli mají od daného uživatele udělené body, kolik mají bodů celkově a kolik mají ohodnocených filmů. Do této statistiky se započítávají pouze uživatelé s minimálně jedním přihlášením do měsíce. Aktualizace statistik probíhá jednou denně.

Pokud si uživatel prochází jednotlivé filmy, je každému věnována samostatná stránka, na které uvidí plakát filmu a základní informace o tomto filmu. Základní informace jsou český, slovenský a anglický název filmu, dále informace o žánrech filmu, státech ve kterých byl film natáčen, roku natočení filmu a délce trvání filmu. Dalšími důležitými informacemi o filmu je režisér natáčející tento film a také herci hrající v tomto filmu.
Na stránce dále najdeme procentuální úspěšnost filmu u uživatelů těchto stránek a také komentáře uživatelů k tomuto filmu. Stránka obsahuje i informace o tom, kdy bude nebo byl daný film uváděn v českých a slovenských kinech a kdy bude nebo začal být vydáván na DVD. Uživatel zde také najde odkazy například na zajímavosti o filmu, na

galerii obrázků, videí z filmu, na stránky IMDb a na stránky s možností zakoupení filmu. Některé výše zmiňované údaje nemusí být uvedeny u každého filmu.

Uživatel také může prohlížet profily herců a režisérů, které mohou obsahovat jméno herce/režiséra, datum narození, místo narození, jeho biografii a informace o filmech, které režíroval nebo ve kterých hrál.

2.1.3 FDb(Filmová databáze)

Filmová databáze [5] je konkurenční český filmový server stránek ČSFD. Stránky FDb, jejich obsah a funkčnost je velmi podobná stránkám ČSFD. Stránky jsou spravovány společností Filmová databáze s.r.o. a založeny byly v roce 2003. FDb není tak rozšířená jako ČSFD, což dokládá počet registrovaných uživatelů, kterých je 72622. Filmová databáze také pokulhává v počtu titulů(filmů) uvedených na svých stránkách, je jich 54979. FDb má však lépe zpracovány osoby podílející se na vzniku filmu. U jednotlivých herců se snaží uvádět jakou postavu ve filmu ztvárnili. Dále se také snaží uvádět osoby spojené s námětem, scénářem, hudbou, kostýmy, zvukem, produkcí, výpravou, kameramany, zvukaře a další. Filmová databáze má také lepší hodnocení, jelikož svou stupnici rozdělila od 1 do 10 oproti ČSFD, která má stupnici pouze 0 až 5. FDb se zaměřuje na kompletní průřez českou a slovenskou kinematografií od němých filmů až po současnost, na stránkách naleznete téměř všechny filmy, které u nás byly k vidění od roku 1988. Další rozdíl mezi FDb a ČSFD je, že na FDb může každý zaregistrovaný uživatel napsat komentář k filmu. Tyto komentáře se pak zobrazují podle počtu bodů (prestiže) uživatele, který je napsal. Další nabízené informace a funkcionalita jsou podobné u obou serverů a jsou popsány v kapitole 2.1.2 s názvem ČSFD.

3 Webový robot

Webový robot [6] je počítačový program, který pro svého majitele opakovaně vykonává nějakou rutinní činnost na internetu - obvykle sbírá data, odesílá a zpracovává požadavky na služby vzdálených serverů. Může se jednat o vyhledávací roboty, spamovací roboty, roboty pro správu a údržbu nebo botnety.

Vyhledávací roboti jsou například využiti v internetových vyhledávačích. Tito roboti procházejí jednotlivé webové stránky a hledají na nich odkazy na další stránky, indexují již zpracované stránky a umožňují jejich následné procházení.

Spamboti jsou roboti, kteří mohou do stránek vkládat data a provádějí tak komentářový spam nebo mohou ze stránek získávat data například e-mailové adresy, které jsou pak využity pro rozesílání spamu (nevyžádané pošty).

Roboti pro správu a údržbu webu jsou například využívání Wikipedií pro údržbu mezijazykových odkazů, vytváření nových přesměrování nebo odstraňování nefunkčních externích odkazů.

Botnety často staví dnešní počítačové viry, botnety jsou sítě propojených botů na počítačích napadených virem a čekající na příkazy majitele, ke splnění těchto úkolů využívají napadené počítače.

3.1 Vyhledávací robot

Vyhledávací robot [7] použitý v internetových vyhledávačích neustále automatizovaně prochází velké množství internetových stránek, propojených hypertextovými odkazy. Vyhledávací robot začíná se seznamem URL adres stránek, které musí projít. Robot vyhledává odkazy na aktuálně procházených stránkách a přidává zjištěné adresy do seznamu URL adres, které je ještě třeba projít. Robot si vytváří kopii každé navštívené stránky, která slouží pro vytváření indexu všech slov, která vyhledávací robot nalezne.

Všechny stránky však nemusí být nalezeny, mohou být vyhledávačem přehlédnuty a to hlavně z těchto důvodů:

- stránky nejsou prostřednictvím odkazů dobře propojeny s ostatními(např.čistě grafická navigace pomocí Flashového menu bez hypertextových popisů)
- stránky byly v době procházení nepřístupné nebo se na nich vyskytovala chyba
- stránky byly přidány až po projití vyhledávače

Dalším problémem může být neschopnost vyhledávacího robota vytvořit kopii webové stránky. Tento stav může nastat v důsledku nevhodného obsahu a struktury webových stránek. Může nastat problém se stránkami s dynamickým obsahem (např. použití Flash, JavaScriptu nebo dynamicky generovaného URL). Tuto chybu lze zkontrolovat pomocí textové podoby stránky.

Další překážkou správného indexování může být samotná adresa stránky. Správně optimalizované URL adresy by měly být stálé, stručné, sémanticky správné a neměly by obsahovat speciální znaky vyžadované programovým kódem stránek.

Jinou možností je, že vlastník stránky nechce, aby daná stránka byla zpracována vyhledávacím robotem, a tak použije protokol pro zakázání přístupu robotům.

Jak dosáhnout co nejlepších výsledků při vyhledávání našich stránek a jak se vyvarovat výše zmíněným chybám nám ukazuje SEO.

3.1.1 SEO(Search Engine Optimization)

SEO [8] je optimalizace webových stránek pro internetové vyhledávače. Jedná se o souhrn metod, technických prostředků a zásad umožňující správné zpracování internetovým vyhledávačem. Hlavní metody SEO jsou:

- kvalitní a unikátní obsah
- správné používání doporučených sémantických značek
- správný titulek webové stránky
- používání description, keywords a dalších meta informací o webové stránce
- budování zpětných odkazů
- přívětivé URL adresy

3.2 Robot

V této práci je využit speciální případ vyhledávacího robota, který slouží k vyhledávání a ukládání potřebných dat ze serveru ČSFD do databáze. Robot vytvořen pro tuto diplomovou práci, lze použít jen pro sběr daných dat na daných stránkách s přesně definovanými prvky. To znamená, že tento robot nelze využít pro stahování jiných dat nebo stahování ze stránek se změněným rozpoložením prvků na stránce. Pokud by došlo ke změně stránek nebo dat, musel by se program náležitě upravit.

4 Podobnostní vyhledávání, metriky

Studií podobnosti [9] se lidstvo zabývá již po staletí, dotýká se našeho rozhodování, pochopení a úvah zásadním způsobem. Až s rozvojem techniky se začalala podobnost využívat v různých oblastech souvisejícís informatikou, jako rozpoznávání, dolování dat a databázové systémy. V dnešní době jsou skoro veškeré informace prezentovány v digitální podobě. Tradiční způsob vyhledávání založený na shodě není použitelný pro multimedia, které tvoří většinu internetových dat. Vhodným řešením je používat koncept podobnosti, který podle zadaných parametrů vyhledává podobná data. Problematika specifikace podobnosti (blízkost dat) je důležitou oblastí, jelikož volba podobnostní funkce ovlivňuje jak kvalitu výsledku, tak rychlost nalezení výsledku. Zvolíme-li podobnost nevhodně, pak prohledávání obrovské kolekce dat v rozumném čase nemusí být možné. V tomto příkladě budeme brát podobnost jako funkci, která pro pár objektů vrátí jedno reálné číslo. Definice podobnosti $S : U \times U \longmapsto R$. Tato funkce se nazývá párová podobnostní funkce a vrací reálné číslo představující míru podobnosti mezi dvěma vstupními objekty z množiny objektů.

4.1 Metrický prostor

Podobnostní funkce [10] je definována na metrickém prostoru a značí metrickou vzdálenost mezi objekty. O výběru správné metriky rozhoduje typ zpracovávaných dat. Metrický prostor je matematická struktura, pomocí které lze formálním způsobem definovat pojem vzdálenosti.

Definice 4.1 *Metrický prostor je dvojice (M, p), kde M je libovolná neprázdná množina a p je tzv. metrika, což je zobrazení $p : M \times M \to \Re$, které splňuje nasledující vlastnosti pro libovolná $x, y, z \in M$*

- *Vlastnost nezápornosti:* $p(x, y) \geq 0$
- *Vlastnost totožnosti (reflexivita):* $p(x, y) = 0 \Leftrightarrow x = y$
- *Vlastnost symetrie:* $p(x, y) = p(y, x)$
- *Trojúhelníková nerovnost:* $p(x, z) \leq p(x, y) + p(y, z)$

Pokud x a y jsou z množiny reálných čísel a je dána metrika $\rho(x, y) = |x - y|$, pak se jedná o úplný metrický prostor. Trojúhelníková nerovnost je druh tranzitivity, která říká, že jestliže x, y a y, z jsou podobné, pak i x, z jsou podobné.

Pokud funkce nesplňuje všechny vlastnosti metriky, již se nejedná o metriku. Pokud není splněna trojúhelníková nerovnost, jedná se *semimetriku*. Jestliže není splněna podmínka symetrie, jedná se o *quasimetriku*. Pokud není dodržena reflexivita, jedná se o *pseudometriku*. V poslední době se stále více objevují příklady, kdy nelze účinnost metrických funkcí považovat za dostačující. Tak vznikají stále častější požadavky na funkce,

které nejsou metrikami. Hlavním důvodem pak je zvýšení svobody a přesnosti podobnostního modelování, při použití nemetrické podobnosti. Jedny z výhod nesplnění vlastností metriky jsou *robusnost*, robustní funkce je odolná vůči odlehlým objektům, a *lokalita*, místně citlivé funkce mohou ignorovat některé části.

Jak bylo zmíněno výše, je více druhů metrik, proto bude následovat představení nejznámnějších metrik.

4.1.1 Minkowského vzdálenosti

Mezi nejznámější podobnostní funkce patří Minkowského vzdálenosti [9] neboli Lp metriky. Jsou definovány na n-dimensionálních vektorech reálných čísel:

$L_p(x,y) = \left(\sum_{i=1}^{d} |x_i - y_i|^p\right)^{1/p}$

Kde x a y jsou vektory a d je dimenze (velikost) těchto vektorů. Parametr p pak určuje, o kterou metriku se jedná. Pokud bude $p \prec 1$, již se nebude jednat o metriku, protože nebude platit trojúhelníková nerovnost.

Tyto metriky mají široké využití ve vědě, často se používají při porovnávání vektorů zastupujících různé vlastnosti objektů. Některé z těchto funkcí můžeme nalézt pod užívanějšími názvy:

- Manhattanská metrika je označení pro L1 metriku, $L_1(x,y) = \sum_{i=1}^{d} |x_i - y_i|$, parametr p=1
- Euklidovská metrika se říká L2 vzdálenosti a určuje délku úsečky mezi dvěma body, $L_2(x,y) = \sqrt{\sum_{i=1}^{d} |x_i - y_i|^2}$,parametr p = 2
- Maximální vzdálenost $L_\infty(x,y) = \max_{i=1}^{d} |x_i - y_i|$, parametr $p \to \infty$.

4.1.2 Levenshteinova vzdálenost

Levenshteinova vzdálenost [9] neboli Editační vzdálenost určuje podobnost mezi dvěma řetězci znaků x = x1...xn, y = y1...yn. Udává minimální hodnotu operací (vkládání, mazání a nahrazování), které je třeba pro transformování jednoho řetězce na druhý. Při rozdílných cenách za jednotlivé operace bude porušena vlastnost symetrie a nebude se již jednat o metriku.

4.1.3 Earth Mover's Distance (EMD)

EMD je metrika [9], která může být aplikována na různé vícerozměrné struktury. Proto má široké uplatnění, je schopna porovnávat jak textové řetězce, tak i složitější struktury. Znázorňuje, kolik práce musí být uděláno, aby se jeden objekt transformoval ve druhý. Tato vzdálenostní funkce je náročnější na výpočet, než všechny předešlé, avšak v poslední době se těší velké pozornosti. Nechť c_{ij} jsou náklady na transformaci objektu x_i na objekt y_j a f_{ij} je minimální cenový tok mezi x_i a y_j.

Definice 4.2 *Earth Mover's Distance je definována takto*
$EMD(x,y) = \min\left\{\sum_{i=1}^{d}\sum_{j=1}^{d} c_{ij}f_{ij}\right\}$
kde platí

- $f_{ij} \geq 0$
- $\sum_{i=1}^{d} f_{ij} = y_j \forall j = 1,\ldots,d$
- $\sum_{j=1}^{d} f_{ij} = y_i \forall i = 1,\ldots,d$

Earth Mover's Distance je metrikou pokud platí že $c_{ik} \leq c_{ij} + c_{ik} \forall i,j,k$.

4.1.4 Příklady nemetrických funkcí

Již v předchozích třech metrikách je uvedeno, za jakých podmínek přestávají být metrikami. Stejně jako metrik je již v dnešní době mnoho nemetrických funkcí.

- *Cosine meansure & distance* [10] je vhodnou podobnostní funkcí v případech, kdy rozsah vektorů není důležitý, důležité jsou pouze směry daných vektorů. Jedná se o cosínus úhlu mezi dvěma vektory a je definován

  $s_{\cos}(x,y) = \frac{\sum_{i=1}^{d} x_i y_i}{\sqrt{\sum_{i=1}^{d} x_i^2 \cdot \sum_{i=1}^{d} y_i^2}}$

  X a y jsou porovnávané vektory, d je dimenze těchto vektorů.

- *Kullback-Leibler divergence* (KLD) [10] je funkce pro porovnávání histogramů. Je používána jako nepodobnostní funkce určující rozdíl mezi histogramy. Je definována takto

  $\delta_{KLD}(x,y) = \sum_{i=1}^{d} X_i \cdot \log\left(\frac{x_i}{y_i}\right)$

  Kde x a y jsou porovnávané histogramy a d udává dimenzi histogramu.

- *Dynamic Time Warping Distance* (DTW -dynamického borcení časové osy) [11] je funkce pro porovnávání časových řad. Myšlenka algoritmu je taková, že se postupně prochází porovnávané sekvence a upravují se jejich časové osy tak, aby ve výsledku tytéž sekvence s takto upravenými časovými osami měly lépe zarovnaná maxima a minima (tvar obecně).

  Mějme matici M řádu m x n, kde m = $|s1|$, n = $|s2|$, kde s1 a s2 jsou porovnávané posloupnosti. Buňka matice M(i,j) pak odpovídá parciální vzdálenosti $\delta(s1(i), s2(j))$. DTW(s1,s2) je nejkratší cesta v matici (ve smyslu součtu hodnot buněk na cestě). Kde buňky na cestě mají vlastnosti:

  - monotónnost – buňky uspořádány monotónně
  - spojitost – nepřerušená cesta, jednotlivé buňky spolu sousedí
  - hraniční podmínka – první buňka je v matici na souřadnicích (0,0), poslední na souřadnicích (m-1, n-1)

- *Normalized Edit Distance* (NED) [10] je funkce pro porovnávání řeťězců a vrací minimální počet změn, přeměny jednoho řeťězce na druhý. Každá operace může mít různou nezápornou hodnotu.
- *Longest Common Subsequence* (LCS - nejdelší společná podposloupnost) [11] jedná se o hledání nejdelšího společného podřetězce. Jde v podstatě také o hledání cesty v matici. Řetězec x je podřetězcem řetězce y pokud je zde striktně rostoucí posloupnost indexů, takových že existuje spojení mezi symboly z řetězce x a symboly z řetězce y. Příklad LCS(aTCTgAtC, TgCaTAC) se rovná 5, velkými písmeny je vyznačen podřetězec.

5 Popis algoritmu a analýza dat

Práce postupovala v následujících krocích:

1. Analýza dostupných dat na webu filmové sociální sítě ČSFD a návrh vhodné databáze, do které se budou potřebná data ukládat.
2. Sestavení webového robota, který provede sběr potřebných dat z vybraného webu.
3. Nastudování možných metod pro analýzu a vyhledávání v datech stažených webovým robotem.
4. Implementace vybrané metody.
5. Prezentace výsledných dat.

Nejprve bude uvedena analýza dat a datových struktur pro webového robota pro stahování dat ze stránek ČSFD. Následně bude popsána samotná práce programu webového robota. Po této části bude následovat analýza problematiky porovnávání a analýza získaných výsledných dat. A nakonec bude uvedena část popisující funkčnost druhého programu provádějícího zpracování sesbíraných dat webovým robotem. Poslední části je pak samotné zobrazení výsledků.

Pro práci byla vybrána data zveřejněná na stránkách ČSFD. Výběr filmové databáze probíhal mezi ČSFD a FDb. FDb má sice propracovanější informace o filmu, avšak má méně filmů v databázi (FDb-54979, ČSFD-247042). FDb má také méně registrovaných uživatelů (FDb-72622, ČSFD-218752). FDb uvádí, že její databáze obsahuje 710378 hodnocení, ČSFD toto číslo neuvádí, avšak z počtu uživatelů, filmů a známosti stránek je zřejmé, že obsahuje více hodnocení. I přestože má FDb lepší systém ohodnocení filmů, byla vybrána ČSFD pro větší množsví využívaných dat.

5.1 Využité prostředky

Nyní bude následovat seznam použitých prostředků pro vytvoření programové části této diplomové práce.

5.1.1 Programovací jazyk C#

Jedná se o jazyk vytvořený firmou Microsoft. Pro svůj běh tak potřebuje .NET Framework, což je platforma, nad kterou jsou programy spouštěny. C# je objektově orientovaný jazyk, jehož syntaxe vychází z jazyku C++. Je však velmi podobný jazyku Java. Jazyk C# je integrován ve vývojovém prostředí Visual Studio společnosti Microsoft.

5.1.2 .NET Framework

.NET Framework [12] je počítačová platforma usnadňující vývoj aplikací. Běží na počítačích s operačním systémem Microsoft Windows. Součástí platformy .NET Framework je

několik hlavních komponent. První je bázová knihovna tříd (FCL). Druhou komponentou je virtuální exekuční systém (CLR), který řídí a stará se o správné spouštění a běh aplikací. Třetí komponentou je společný typový systém (CTS). Poslední komponentou je společná jazyková specifikace (CLS).

5.1.3 Visual Studio

Pro psaní kódu bylo vybráno vývojové prostředí Microsoft Visual Studio 2008. Je to mocné programátorské prostředí pro tvorbu aplikací. Využívá platformu .NET Framework. Umožnuje vytvářet jak konzolové, tak i grafické desktopové aplikace. Umožňuje také tvorbu webových aplikací. Vestavěné jazyky jsou C#, Visual Basic .NET a C++, další jazyky mohou být přidány jazykovými službami, avšak musí se instalovat zvlášť.

5.1.4 SQL Server

MS SQL Server je relační databázový systém vytvořený firmou Microsoft. Jeho hlavními dotazovými jazyky jsou SQL a T-SQL. Visual Studio sice obsahuje komponentu pro práci a správu databází, která byla nejprve v tomto projektu využita, avšak lepším nástrojem je SQL Server Management Studio, které využívá nakonec i tento projekt. SQL Server Management Studio je grafické rozhraní pro jednoduché vytváření a údržbu databází.

5.1.5 LINQ

Tato diplomová práce bude využívat pro manipulaci s daty v databázi LINQ [13]. Linq je zkratka Language INtegrated Query. Je to integrovaný jazyk pro dotazování. Je to sada rozšíření .NET Frameworku a byl představen v .NET Framework 3.5. Pomocí třídy knihoven rozšiřuje jazykovou syntaxi pro dotazy nad daty jazyků C# 3.0 a Visual Basic 9. LINQ přináší nový způsob pro dotazování nad jakýmikoliv daty, usnadňuje jejich tvorbu, třídění, jejich propojování i vyhledávání v nich. LINQ je obecný nástroj pro práci s daty [14]:

- LINQ to Objects dotazování nad objekty. Pracuje s kolekcemi, které se nacházejí v paměti a inplementují rozhraní IEnumerable$\prec$T$\succ$. Jsou to například pole.
- LINQ to SQL usnadňuje práci s databázemi. Umožňuje dotazy nad databázemi využívajícími rozhraní MS SQL. Příkazy LINQu jsou mapovány na odpovídající příkazy v jazyce SQL. Při získání dat z databáze se pak musí relační data databáze převést na objektová data, se kterými LINQ pracuje. Výhodou LINQu je pak právě objektový pohled na data. LINQ pracuje se servery Microsoft SQL Server 2000 a vyšší.
- LINQ to XML umožňuje práci s XML soubory. Při přístupu k datům nevyužívá DOM (Document Object Model - objektový model dokumentu) ani SAX (Simple API for XML), ale přistupuje k datům plně objektově.
- LINQ to DataSet umožňuje práci s ADO .NET datasety.

- Další implementace LINQ dostupné na internetu LINQ to Hibernate, DbLINQ nebo LINQ to Amazon.

Některá klíčová slova LINQu:

- Select – výběr hodnoty kterou chceme použít
- SelectMany – výběr více hodnot najednou (např. pole)
- Join – spojení více poskytovatelů dat
- GroupBy – rozdělení dat do více skupin podle určitého klíče
- Where – omezení výběru prvků podle specifikované podmínky
- OrderBy, OrderByDescending – specifikace třídění, umožňuje výběr elementu podle kterého se má třídit
- ElementAt – výběr prvku podle udaného indexu
- Count – počet prvků v kolekci
- Union, Intersect, Except – definice množinových operací sjednocení, rozdíl a průnik
- Sum, Min, Max, Average – vrací součet, minimální, maximální či průměrnou hodnotu z dané kolekce
- Reverse – otočí pořadí prvků v kolekci
- Concat – spojí dvě kolekce dohromady
- Take - vypíše maximalně zadaný počet záznamů

Nyní bude uveden příklad práce s LINQ, jedná se odotaz do databáze. Tento ukázkový kód je použit v této práci při získání sta uživatelů s nejvíce podobným vkusem pro daného uživatele. Uživatelé musí mít pět a více společných filmů.

Příklad 5.1
Nejprve je třeba zadat, s jakou databází budeme pracovat. Namapujeme databázi CSFD.

```
- using (var data = new CSFDEntities()) {
```

Nyní můžeme vytvořit samotný dotaz, kde nejprve zadáme, odkud bude brát data. Bude používat veškerá data z tabulky Shoda.

```
- var dotaz2 = (from c in data.Shoda
```

Poté může následovat uvedení podmínek pro vyhledávaná data. Budou vyhledávány jen údaje, kde IDA je rovno *setrideni1* s hodnotou PocSpol větší než 4.

```
- where c.IDA == setrideni1 && c.PocSpol ≻ 4
```

Dále je možné uvést způsob setřídění vyhledaných prvků. Nejprve se setřídí data vzestupně podle Shoda1, dále podle PocSpol sestupně a nakonec vzestupně podle IDB.

- orderby c.Shoda1, c.PocSpol descending, c.IDB

Nakonec se uvede, které atributy chceme získat. Je možné nastavit ještě další vlastnosti uvedené ve výpisu klíčových slov výše. Zajímá nás jen IDB, Shoda1 a PocetSpol. Výpis omezíme pouze na sto prvních záznamů.

- select new c.IDB, c.Shoda1, c.PocSpol).Take(100); }

■

5.2 Analýza webového robota

Stránky ČSFD obsahují spoustu informací. K této práci bude však potřeba získat jen malého množství z nich. Program bude využívat pro hledání potřebných dat fulltextové vyhledávání. Fulltextové vyhledávání bude použito, protože program bude vědět, kde přesně daná data na stránce najde a jak se k ním dostat. Musí procházet textovou podobu stránek a vyhledávat klíčová slova na správných místech. K vyhledávání klíčových slov v textu bude využíváno metody string.IndexOf. Metodě se může zadat, co se má v daném textu hledat, odkud se začne hledat a jak daleko se bude prohledávat. Jestliže nenajde některé z klíčových slov, musí ukončit svou činnost, protože by již nedokázal získat požadovaná data. Pokud se mu podaří dostat až k požadovaným datům, uloží tyto data do databáze.

Pro potřeby dalšího zpracování jsou ukládány základní data o jednotlivých uživatelích, kteří ohodnotili různé filmy. Tyto informace se ukládájí do tabulky *Uzivatel*, v které jsou uloženy ID uživatele, přezdívka uživatele, jeho jméno, bydliště, okres a krátké info o uživateli. Struktura tabulky je vidět v tabulce 1.

Ukládají se i základní informace o jednotlivých ohodnocených filmech. Pro tyto informace byla vytvořena tabulka *Film*, v které jsou uloženy ID filmu, které je převzato z databáze ČSFD, název filmu v češtině, rok natočení a délka filmu. Struktura tabulky je vidět v tabulce 2. Dále se k filmu ukládají i žánry filmu a místa natáčení, tyto data jsou ukládány do tabulek *Zanr* a *Misto*. Struktura těchto 2 tabulek je zobrazena v tabulkách 3 a 4. Dále se ukládají režiséři daného filmu a herci hrající v daném filmu do tabulek *Reziser* a *Herec*. Struktura těchto 2 tabulek je zobrazena v tabulkách 6 a 5.

Posledním ukládaným údajem je hodnocení jednotlivých filmů daným uživatelem. V této tabulce *Hodnotil* je uloženo ID herce, ID filmu a hodnocení filmu. Struktura tabulky je vidět v tabulce 10.

5.2.1 Databáze webového robota

Jedná se o MS SQL databázi, vytvořenou přímo pomocí Microsoft Visual Studia 2008. Z předchozí části 5.2 vyplývá, která data je potřeba ukládat do databáze.

název	datový typ	vlastnost
IDU	int	klíč
Prezdivka	varchar(25)	možno nic nezadat
JmenoU	varchar(80)	možno nic nezadat
Mesto	varchar(50)	možno nic nezadat
Okres	varchar(50)	možno nic nezadat
Info	varchar(50)	možno nic nezadat

Tabulka 1: Tabulka Uzivatel

název	datový typ	vlastnost
IDF	int	klíč
NazevF	varchar(100)	možno nic nezadat
Rok	int	možno nic nezadat
Delka	varchar(15)	možno nic nezadat

Tabulka 2: Tabulka Film

5.2.1.1 Datová analýza Jedná se o proces rozpozvávání reálných objektů, jejich vlastností a jejich vazeb s dalšími objekty. Jeho výsledkem je pak model datových struktur projektu. Výsledky datové analýzy jsou pak prezentovány těmito prostředky [15] :

- lineární zápis typů entit -specifikace entitních typů a vztahů mezi těmito entitními typy
- lineární zápis typů vazeb -uvedení vazeb mezi entitami
- E-R diagram -grafické znázornění entit a jejich vazeb
- popis a specifikace atributů
- datový slovník -slovní formalizovaný popis dat

Následující text, tabulky a obrázek uvádí datovou anlalýzu webového robota.

Databáze obsahuje 11 tabulek, z toho 5 vazebních. Jsou to tyto tabulky *Uzivatel*, *Film*, *Zanr*, *Misto*, *Reziser*, *Herec*, *FilmHrali*, *FilmNatocen*, *FilmZanr*, *Rezie* a *Hodnotil*. Strukturu databáze můžete vidět na obrázku 1. Jednotlivé struktury všech tabulek databáze můžete vidět v níže zobrazených tabulkách.

Tabulka *Uzivatel* má hodnotu klíče *IDU* generovanou automaticky databází. U atributů JmenoU, Mesto, Okres a Info jsou ponechány velikosti v trochu větším rozsahu kvůli speciálním symbolům, které mohou uživatelé do svých profilů zadat a které jsou v textové podobě stránky reprezentovány více znaky.

V tabulce *Film* je hodnota klíče *IDF* shodná s hodnotou id daného filmu v databázi ČSFD. Atribut délka je z důvodu různých formátů času ponechán v textové podobě. Některé případy zadání časů 125, 2x100, 49+48, 65 + 75 = 140.

Tabulky *Zanr* a *Misto* mají své klíče *IDZ* a *IDM* generovány automaticky databází.

název	datový typ	vlastnost
IDZ	int	klíč
NazevZ	varchar(15)	možno nic nezadat

Tabulka 3: Tabulka Zanr

název	datový typ	vlastnost
IDM	int	klíč
Zeme	varchar(30)	možno nic nezadat

Tabulka 4: Tabulka Misto

V tabulkách *Reziser* a *Herec* jsou hodnoty klíčů *IDR* a *IDH* shodné s hodnotami id daných režisérů a herců v databázi ČSFD.

Vazební tabulky jsou *FilmHrali*, *FilmNatocen*, *FilmZanr*, *Rezie* a *Hodnotil*. První čtyři tabulky pouze propojují tabulky bez dalších informací o vazbě mezi nimi.

Tabulka *FilmHrali* je vazební tabulkou mezi *Film* a *Herec*, ze kterých také přebírá své dva atributy. Z tabulky *Film* IDF a z tabulky *Herec* IDH. Oba atributy dohromady tvoří složený klíč tabulky. Tabulka zastupuje vazbu *herec hraje ve filmu*.

Tabulka *FilmNatocen* je vazební tabulkou mezi *Film* a *Misto*, ze kterých také přebírá své dva atributy. Z tabulky *Film* IDF a z tabulky *Misto* IDM. Oba atributy dohromady tvoří složený klíč tabulky. Tabulka zastupuje vazbu *film byl natáčen na místě*.

Tabulka *FilmZanr* je vazební tabulkou mezi *Film* a *Zanr*, ze kterých také přebírá své dva atributy. Z tabulky *Film* IDF a z tabulky *Zanr* IDZ. Oba atributy dohromady tvoří složený klíč tabulky. Tabulka zastupuje vazbu *film má žánr*.

Tabulka *Rezie* je vazební tabulkou mezi *Film* a *Reziser*, ze kterých také přebírá své dva atributy. Z tabulky *Film* IDF a z tabulky *Reziser* IDR. Oba atributy dohromady tvoří složený klíč tabulky. Tabulka zastupuje vazbu *film byl natáčen režisérem*.

Poslední vazební tabulka je propojení mezi tabulkou *Film* a *Uzivatel*. Její složený klíč tvoří převzaté atributy IDU z *Uzivatel* a IDF z *Film*. Posledním atributem v tabulce je Hodnoceni. Tabulka ralizuje vazbu *uživatel hodnotil film*

5.2.1.2 Funkční analýza Tato část analýzy navazuje na datovou analýzu, a má za úkol popsat veškeré operace prováděné nad dříve specifikovanými daty. Jedná se o operace vkládání, mazání, editaci, výpočet nad daty atd. Funkční analýza může užít pro zobrazení a popis operací nad daty tyto prostředky:

název	datový typ	vlastnost
IDH	int	klíč
JmenoH	varchar(60)	možno nic nezadat

Tabulka 5: Tabulka Herec

název	datový typ	vlastnost
IDR	int	klíč
JmenoR	varchar(60)	možno nic nezadat

Tabulka 6: Tabulka Reziser

název	datový typ	vlastnost	informace
IDF	int	složený klíč	atribut z tabulky Film
IDH	int	složený klíč	atribut z tabulky Herec

Tabulka 7: Vazební tabulka FilmHrali

název	datový typ	vlastnost	informace
IDF	int	složený klíč	atribut z tabulky Film
IDM	int	složený klíč	atribut z tabulky Misto

Tabulka 8: Vazební tabulka FilmNatocen

název	datový typ	vlastnost	informace
IDF	int	složený klíč	atribut z tabulky Film
IDZ	int	složený klíč	atribut z tabulky Zanr

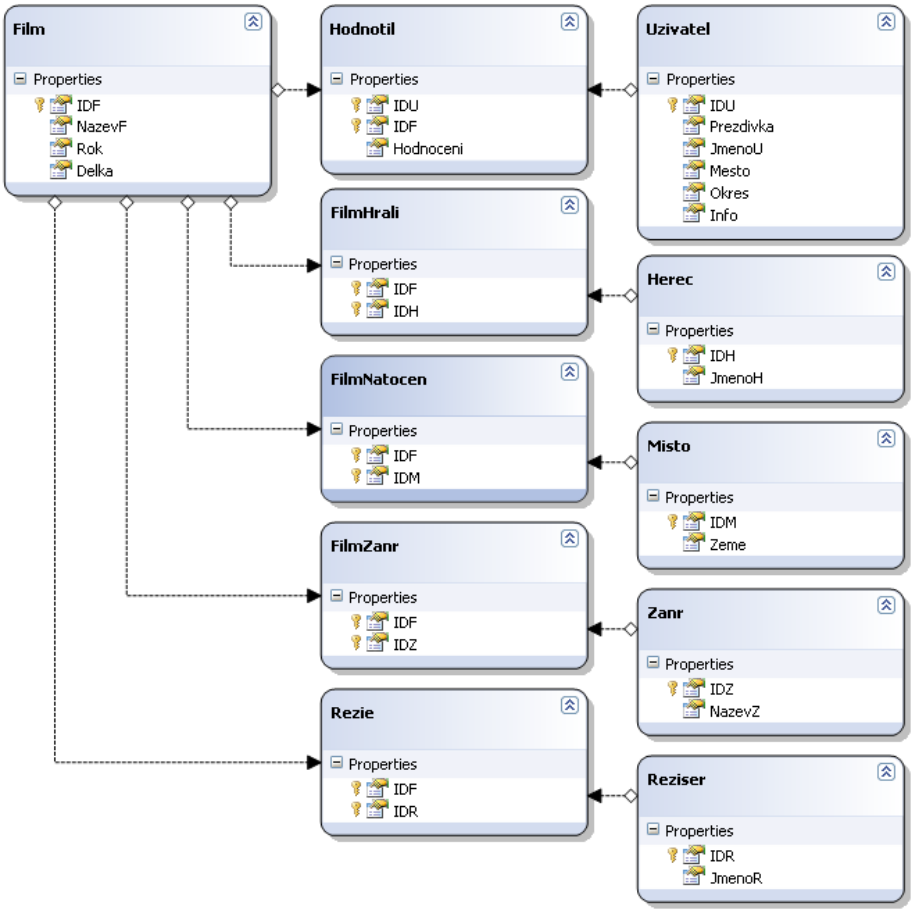
Tabulka 9: Vazební tabulka FilmZanr

název	datový typ	vlastnost	informace
IDU	int	složený klíč	atribut z tabulky Uzivatel
IDF	int	složený klíč	atribut z tabulky Film
Hodnoceni	int	možno nic nezadat	

Tabulka 10: Vazební tabulka Hodnotil

název	datový typ	vlastnost	informace
IDF	int	složený klíč	atribut z tabulky Film
IDR	int	složený klíč	atribut z tabulky Reziser

Tabulka 11: Vazební tabulka Rezie



Obrázek 1: Obrázek schématu tabulek

- diagram datových toků (DFD) -grafický prostředek pro zobrazení funkčního modelu projektu
- minispecifikace -popis práce elementární funkcí
- diagram aktivit -popis dynamických aspektů projektu
- datový slovník -formální popis datových toků

Z pohledu funkční analýzy je tento program velmi jednoduchý. Z pohledu uživatele má jedinou elementární finkci a to stahování dat. Tato funkce je sice obsáhlá, avšak již dále nedělitelná. Měla by proběhnout celá, po přerušení musí začít od začátku. Nebude zde tedy uvedena žádná z částí funkční analýzy. Funkční analýzu naleznete v příloze.

5.3 program SberacDat(webový robot)

Algoritmus 1: Sběr dat ze stránek ČSFD

```
   Data: Data potřebná pro přecházení mezi stránkami
   Result: Naplněná databáze sebranými daty při procházení stránek ČSFD
1  Načtení úvodní stránky ČSFD;
2  Nalezení a otevření odkazu Uživatele;
3  Nalezení a otevření odkazu Podrobnější přehledy uživatelů;
4  Zjištění počtů stránek s uživateli;
5  for Procházej postupně všechny stránky do
6      for Procházej postupně všechny uživatele na aktuální stránce do
7          Načtení stránky aktuálního uživatele;
8          Vyhledání a uložení dat o uživateli;
9          Nalezení a otevření odkazu Hodnocení u aktuálního uživatele;
10         Zjištění počtů stránek s hodnoceními;
11         for Procházej postupně všechny stránky s hodnoceními do
12             for Procházej postupně všechny filmy na aktuální stránce s hodnoceními do
13                 if Pokud film ještě nebyl zpracován then
14                     Načtení stránky aktuálního filmu;
15                     Vyhledání a uložení dat o filmu;
16                     Vyhledání, zpracování a uložení režisérů filmu;
17                     Vyhledání, zpracování a uložení herců hrajících ve filmu;
18                 Vyhledání a uložení ohodnocení aktuálního filmu aktuálním
                   uživatelem;
19 Ukončení práce programu;
```

Řádek 1 Provede se načtení zadané internetové adresy www.csfd.cz. Načítání všech stránek probíhá pomocí funkce string nacteniStranky(string adrPrac), Návratovou hodnotou je text načítané stránky. Proměnná adresa obsahuje požadovanou stránku k načtení.

Řádek 2 Prohledává se textová podoba stránky www.csfd.cz a hledá se zda obsahuje text `>Uživatelé<`.Vyhledávání tohoto textu probíhá pomocí funkce *vyhledavani*, tato funkce vrací pole znaků obsahující adresu další stránky. Pokud je daný text nalezen, pokusím se načíst pomocí zjištěné adresy novou stránku.

Řádek 3 Prohledává se textová podoba stránky načtené v předešlém kroku a hledá se zda obsahuje text `>Podrobnější přehledy uživatelů<`. Vyhledávání tohoto textu probíhá pomocí funkce *vyhledavani*. Pokud je daný text nalezen, pokusím se načíst pomocí zjištěné adresy novou stránku.

Řádek 4 Prohledává se textová podoba stránky načtené v předešlém kroku. Z této stránky zjišťuji počet stránek s výpisem uživatelů, uživatelé jsou vypisováni na jednotlivé stránky po stovkách. Pro tato vyhledávání používám funkci *hledani*, která vrací pozici hledaného textu.

Řádek 5 Začíná postupné načítání jednotlivých stránek s uživateli, kde se jednotlivé adresy stránek liší pouze v čísle, kterou stovku uživatelů zobrazují.

Řádek 6 Vyhledávání jednotlivých uživatelů na stránce. Uživatelé jsou vyhledávání v textu pomocí svého pořadového čísla. Hledání tohoto textu provádí funkce *vyhledavaniOsob*, která vrací pole znaků s adresou stránky daného uživatele.

Řádek 7 Načtení stránky o uživateli. Provádí se zde kontrola, zda již nebyl zpracován poslední uživatel. Kontrola je nutná, protože program se snaží vždy projít všech 100 uživatelů na dané stránce, na poslední stránce jich však tolik patrně nebude. Dále se zde provádí kontrola, zda byla daná stránka načtena. Pokud nebyla načtena, tak se přeskočí zpracování daného uživatele a přejde se na krok 6 načtení dalšího uživatele.

Řádek 8 Vyhledání informací o uživateli a jejich následné uložení do databáze. Postupně se prochází text stránky a hledají se klíčová slova pro dané informace. Nejprve se hledá přezdívka. Vyhledá se část textu s informacemi o uživateli. Zpracuje se přezdívka a pojkračuje se vyhledáním jména uživatele, které se následně uloží do pole znaků. Pokračuje se vyhledáním města, okresu a informace o uživateli. Tyto údaje nemusí být v textu obsaženy a proto je nutná kontrola. Dále se provádí ověření, zda již není uživatel uložen, aby nedocházelo k jejich opakovanému ukládání, po novém spuštění programu s již části naplněnou databází. Do tabulky *Uzivatel* v databázi ukládáme výše zjištěné informace a to Prezdivka, JmenoU, Mesto, Okres a Info. Klíčová hodnota IDU je generována přímo databází.

Řádek 9 Prohledává se textová podoba stránky uživatele a hledá se, zda obsahuje text `>Hodnocení</a>`.Vyhledávání tohoto textu probíhá pomocí funkce vyhledavani viz výše. Pokud je daný text nalezen, pokusím se načíst pomocí zjištěné adresy novou stránku.

Řádek 10 Prohledává se textová podoba stránky s hodnocením u daného uživatele. Z této stránky zjišťuji počet stránek s výpisem hodnocených filmů daným uživatelem. Hodnocení jsou vypisovány na jednotlivé stránky po stovkách. Pro tyto vyhledávání používám funkci hledani viz výše.

Řádek 11 Začíná postupné načítání jednotlivých stránek s uživatelem ohodnocenými filmy, kde se jednotlivé adresy stránek liší pouze v čísle, kterou stovku hodnocení uživatele zobrazují.

Řádek 12 Vyhledávání jednotlivých hodnocených filmů na stránce. Filmy jsou vyhledávány v textu pomocí odkazu na svou stránku. Při každém filmu hledám tento odkaz tolikrát, kolikatý je daný film v pořadí výpisu na dané stránce. Hledání celé adresy provádí pak funkce string infoFilmu(string html, int pozice1, string hledano, string adresa), která vrací text s adresou stránky daného filmu.

Dále se provede výjmutí id filmu z adresy filmu pomocí funkce *zjistiIDFRH*, která vrací id fimlu.

Pokud není id filmu nalezeno, daný film se dále nezpracovává. Pokud je id filmu získáno, je zjišťováno, zda už není tento film v databázi. Pokud již je v databázi, je jeho zpracování přeskočeno a přejde se na část zapsání hodnocení.

Řádek 13 Načtení stránky o filmu. Provádí se zde kontrola, zda byla daná stránka správně načtena, pokud ne, přeskočí se zpracování daného filmu a přejde se na krok 12 načtení dalšího filmu.

Řádek 14 Vyhledání informací o filmu a jejich následné uložení do databáze. Postupně se prochází text stránky a hledají se klíčová slova pro dané informace. Nejprve se hledá název filmu. Pokud však není ukončovací symbol názvu filmu nalezen nebo je dané dílo na konci označeno jako `(Divadlo)`, `(TV pořad)` ,`(TV seriál)`, pak není tento film vůbec zpracován a přejde se na krok 12 načtení dalšího filmu. V opačném případě se načte název filmu do pole znaků.

Nyní se vyhledává rok natočení filmu a délka filmu. Zde se musí provést několik testů, protože rok natočení je vypsán společně s místem natočení a délkou filmu. Může se stát, že nebude zadána některá z těchto hodnot, nebo může nastat případ, že nebude zadána ani jedna informace z těchto tří hodnot. Testy se provádějí pro možnosti, že byl zadán pouze jeden údaj ze tří, nebo dva údaje, nebo všechny tři. Nejprve se testuje, zda lze daný text převést na číslo, pokud ano, jedná se o rok natočení filmu, pokud ne, testuje se, zda se jedná o délku filmu pomocí vyhledání textu `␣min` a pokud nebyla splněna ani tato podmínka, jedná se o zemi, kde byl film natáčen.

Po zjištění a uložení dat do správných proměnných, se provede zápis filmu a informací o něm do databáze. Do tabulky *Film* v databázi ukládáme výše zjištěné informace a to IDF, NazevF, Rok a Delka.

Dále je zpracován žánr filmu, jeden film může mít více žánrů. Proto se hledání žánru provádí v cyklu a hledá se vždy další oddělovač mezi jednotlivými žánry. Žánry jsou v tomto cyklu jednotlivě zpracovávány. Před zápisem do databáze se testuje, zda již daný žánr není v tabulce *Zanr*. Poté se provádí kontrola, zda již daný žánr nebyl zapsán k právě zpracovávanému filmu v tabulce *FilmZanr*, pokud ne, je zapsán do této tabulky. Do tabulky *FilmZanr* se zapisují IDF a IDZ.

Po zpracování žánru následuje zpracování míst natáčení filmu, jeden film může mít více zemí, v kterých byl natáčen. Proto se hledání míst provádí v cyklu a hledá se vždy další oddělovač mezi jednotlivými místy. Země jsou v tomto cyklu jednotilvě zpracovávány. Nejprve se testuje, zda již daná země nebyla v tabulce *Misto* zapsána, pokud ne, je země zapsána do tabulky. Nyní se provádí kontrola, zda již daná země nebyla zapsána k právě zpracovávanému filmu v tabulce *FilmNatocen*, pokud né, je zapsána do této tabulky. Do tabulky *FilmNatocen* se zapisují IDF a IDM.

Řádek 15 Vyhledání a zpracování režisérů filmů. Nejprve se zkouší, zda je vůbec některý režisér k danému filmu přiřazen. Pokud jsou režiséři vypsáni, provádí se jejich zpracování v cyklu, z důvodu možnosti více režisérů u jednoho filmu. Nejprve se zjistí z odkazu na stránku režiséra jeho id pomocí funkce zjistiIDFRH viz výše. Následně je vypsáno a uloženo do proměnné jméno režiséra. Pak se provádí kontrola, zda již je režisér uložen v databázi v tabulce *Reziser*. Pokud režisér zatím v databázi není, je do tabulky *Reziser* uložen. Následně se provede další kontrola a to, zda již daný režisér není přiřazen k aktuálně zpracovávanému filmu v tabulce *Rezie*. Pokud odpovídající záznam v tabulce není, jsou do tabulky uloženy IDF a IDR. Pokud to již byl poslední režisér, bude se provádět další část programu, jinak se program vrátí na začátek cyklu a zpracovává dalšího režiséra.

Řádek 16 Vyhledání a zpracování herců hrajících ve filmu. Není-li nalezen začátek části s herci, nejsou herci k filmu přiřazeni a jejich zpracování se neprovádí. Jinak se provádí zpracování herců v cyklu, protože ve filmu hraje vždy více herců. V cyklu se nejprve provede kontrola, zda je nyní zpracováván již poslední herec. Poté se vypíše z odkazu na stránku o herci jeho id, následně je vypsáno a uloženo do proměnné jméno herce. Před uložením se provádí kontrola, zda již není herec uložen v databázi v tabulce *Herec*. Následně se provede další kontrola a to zda již daný herec není přiřazen k aktuálně zpracovávanému filmu v tabulce *FilmHrali*. Pokud odpovídající záznam v tabulce *FilmHrali* není, jsou do tabulky uloženy IDF a IDH. Pokud to již byl poslední herec, ukončí se zpracovávání filmu a začne se provádět uložení ohodnocení filmu, jinak se program vrátí na začátek cyklu a zpracovává dalšího herce.

Řádek 17 Vyhledání a zpracování ohodnocení filmu daným uživatelem. Program zjistí, jak byl film ohodnocen pomocí funkce pocetHvezd(string html, int pozice1), která vrací počet hvězd hodnocení (0-5). Po zjištění počtu hvězd, se provede kontrola, zda již údaj se stejným IDF a IDU v tabulce Hodnotil není. Tento krok je zde ponechán pro případ běhu programu s již z části naplněnou databází. Pokud takový údaj v tabulce *Hodnotil* neexistuje, je do ní uloženo IDU, IDF a Hodnoceni. Tímto je zpracování filmu zcela dokončeno a je zpracováván další film uživatele, popřípadě další uživatel, nebo pokud již byli zpracováni všichni uživatelé i jejich filmy, je stahování dat ukončeno.

5.4 Analýza programu pro zpracování dat

Hlavní částí programu pro zpracování dat je způsob porovnání stažených dat, uložených v databázi, ze stránek ČSFD. Nad daty staženými webovým robotem je možné provést vícero porovnání, kdy můžeme zohledňovat data zjištěná z profilů jednotlivých uživatelů, kdy je k dispozici přezdívka, místo bydliště a krátké info. Nebo by bylo možné využít další stažené informace o filmu, jako jsou herci, režiséři, žánr a místa natáčení. Avšak všechny tyto data nemusí být na stránkách ČSFD obsažena a i jejich vyhodnocování by bylo obtížné. Proto tato práce využívá pouze porovnávání zaměřená na ohodnocení filmů jednotlivými uživateli.

Definice 5.1 *Zvolená podobnost je založena na hodnocení filmů jednotlivými uživateli, kdy bere v potaz pouze společné filmy obou porovnávaných uživatelů. Sčítá rozdíly jednotlivých hodnocení*

a výsledné číslo pak podělí počtem společných filmů obou uživatelů. Podobnost je reálné číslo a má stupnici 0-5, kde 0 udává shodné hodnocení všech společných filmů. Hodnota 5 udává zcela opačné názory na kvalitu společných filmů. Pokud nemají dva porovnavaní uživatelé společné filmy, je výsledek, že tito uživatele nemají podobný vkus.

$Podobnost(U1, U2) = (\sum_{i=1}^{s} |U1_i - U2_i|)/s, s \geq 1$
$Podobnost(U1, U2) = 5, s = 0$

- $Podobnost(U1, U2) \in \langle 0, 5 \rangle$ *kde* $U1aU2 \in U$, *U je množina všech uživatelů*
- s *udává počet společných filmů mezi U1 a U2*
- $U1_i$ *je hodnocení i-tého společného filmu prvním uživatelem a* $U2_i$ *je hodnocení i-tého společného filmu druhým uživatelem*

Mohlo být zvoleno ještě jednodušší porovnání a to provést pouze procentuální výpočet shodnosti hodnocení společných filmů, kde by nás nezajímaly velikosti rozdílů v hodnocení u společných filmů. Takovéto porovnání je použito a výsledky prezentovány na stránkách ČSFD u jednotlivých uživatelů v sekci spřízněné duše. Avšak jsou tam ještě omezující prvky, jako uživatel musí být aktivní a mít určitý počet bodů. Na stránkách jsou také zobrazeny shody podle deseti nejoblíbenějších režisérů, herců a hereček, toto porovnání je však velice zavádějící, protože hodně uživatelů bude mít stejný počet shodných režisérů, herců a hereček. Navíc pro takovéto porovnávání nestahuje webový robot data, jelikož je uvádějí jen někteří uživatelé. Víc o použitých porovnáních na stránkách ČSFD v kapitole 2.1.2 ČSFD.

Obecně by bylo výhodné využít pro porovnávání nějakou metriku. Avšak v tomto konkrétním případě to není možné, jelikož data porovnávaná tímto programem nesplňují všechny podmínky metriky. Nesplňují pravidlo tranzitivity.

Může např. nastat tato situace U1{1,2}, U2{1,7} a U3{5,7}, kde *U* jsou uživatelé a čísla jsou filmy, které daní uživatelé hodnotili. Zde platí, že U1 je v relaci s U2 a U2 je v relaci s U3, avšak U1 v relaci s U3 není.

Pro první algoritmus je použit jednoduchý postup, kdy se načtou data dvou porovnávaných uživatelů. Načtená data jsou data z tabulky *Hodnotil*, která je setříděna podle *IDU*(id uživatele). Porovnání se pak provádí hledáním společných filmů obou uživatelů a sčítáním rozdílu mezi jejich hodnocením daného filmu. Po porovnání všech společných filmů je výsledný rozdíl podělen počtem společných filmů. Výsledné číslo je pak v rozmezí 0 až 5 a udává shodu hodnocení obou uživatelů, kde 0 udává shodné hodnocení u všech společných filmů a 5 udává zcela opačné názory na kvalitu společných filmů.

Pro práci druhého algoritmu je potřeba nejprve přetřídit data z tabulky *Hodnotil* tak, aby byla data setříděna podle *IDF*(id filmu). Po setřídění dat se může provádět porovnávání, kdy se nejprve načtou informace o uživateli, kterého chceme porovnat se všemi ostatními. Data o uživateli se načítají z tabulky *Hodnotil*. Nyní se prochází postupně jednotlivé filmy hodnocené daným uživatelem a k danému filmu se z nově vytvořené tabulky zjišťuje, kteří uživatelé také hodnotili daný film. U všech uživatelů se budou pamatovat rozdíly v hodnocení a počet společných filmů s původním uživatelem, což je

název	datový typ	vlastnost	informace
IDA	int	složený klíč	atribut z tabulky Uzivatel
IDB	int	složený klíč	atribut z tabulky Uzivatel
Shoda	float	možno nic nezadat	
PocSpol	int	možno nic nezadat	

Tabulka 12: Tabulka Shoda

uživatel vybraný pro porovnání se všemi ostatními. Před uložením se pak celkový rozdíl u každého uživatele podělí počtem společných filmů. Vznikne tedy stejné ohodnocení jako u předchozí možnosti, popsané v předchozím odstavci.

Práce obou algoritmů bude podrobněji popsána dále.

5.4.1 Databáze programu pro zpracování dat

I tento program využívá lokální databázi na MS SQL Serveru 2008. Oproti webovému robotu bude potřebovat program pro zpracování dat ještě tabulku pro ukládání výsledků porovnání a již dříve zmíněnou tabulku pro setřídění údajů tabulky *Hodnotil* podle id filmu.

5.4.1.1 Datová analýza Datová analýza tabulek naplněných webovým robotem byla provedena u datové analýzy webového robota. Jak bylo uvedeno výše potřebujeme další dvě tabulky. Tabulka pro uložení výsledků se nazývá *Shoda* a obsahuje id dvou porovnávaných uživatelů *IDA* a *IDB*. Tyto dva atributy tvoří složený klíč a jsou převzaty z tabulky *Uzivatel*. Dále tabulka obsahuje výsledné číslo *Shoda*, určující shodu daných dvou uživatelů. Posledním údajem je počet společných filmů *PocSpol* daných dvou uživatelů. Počet společných filmů má sloužit při zobrazování výsledků pro odfiltrování náhodných shod s malým počtem stejných filmů hodnocených oběma uživateli. Strukturu tabulky *Shoda* je možné vidět zde 12.

Tabulka *Shoda* nebude obsahovat z důvodů úspory místa žádná nadbytečná data. Pokud dva daní uživatelé nemají žádný společný film, nebude se záznam ukládat do databáze. Při výpisu, pokud nebude záznam v databázi nalezen, bude program vědět, že daní uživatelé nemají společný film, a ohodnotí tuto podobnost nejhorším výsledkem (číslem 5). Dále také nebude ukládáno opačné pořadí uživatelů v porovnání. Pro vypsání všech podobností k danému uživateli nám nestačí vyhledávat id daného uživatele v prvním indexovaném záznamu(IDA) v tabulce. Musíme také uživatelovo id hledat v druhém záznamu (IDB), pro vyhledání jeho podobnosti s uživateli s nižším id.

SetrideneFilmy je název tabulky pro uložení setříděných dat podle id filmu z tabulky *Hodnotil*. Tabulka tedy bude obsahovat složený klíč *IDF* a *IDU* a atribut *Hodnoceni*. Struktura je zobrazena v tabulce 13.

Pro druhý algoritmus je potřeba vytvořit pomocné pole. Pole má strukturu dvou integeru, kde jeden slouží pro uložení celkového rozdílu v hodnocení mezi dvěma uživateli. Druhý slouží pro uložení počtu společných filmů. Velikost pole je rovna počtu uživatelů.

název	datový typ	vlastnost	informace
IDF	int	složený klíč	atribut z tabulky Film
IDU	int	složený klíč	atribut z tabulky Uzivatel
Hodnoceni	int	možno nic nezadat	

Tabulka 13: Tabulka SetrideneFilmy

každý sloupec odpovídá jednomu uživateli	uživatel č.1	uživatel č.2
celkový rozdíl ze všech společných filmů	0	12
počet společných filmů	0	25

Tabulka 14: Pomocné pole na ukládání počtů společných filmů pro druhou metodu

Hodnota indexu pole plus jedna značí id uživatele. Schéma pomocného pole znázorňuje tabulka 14, kde se ke každému uživateli, reprezentovaným indexem pole, ukládá celkový rozdíl hodnocení a počet společných filmů.

5.4.1.2 Funkční analýza Tento program má tři elementární funkce. Dvě funkce jsou jsou algoritmy výpočtu podobnosti a třetí funkce je funkce pro setřídění tabulky *Hodnotil* podle IDF. Funkční analýza tohoto programu bude uvedena v přílohách.

5.5 Program TrideniDat(program pro zpracování dat)

Algoritmus 2: První metoda pro porovnávání

```
   Data: Databáze s daty sebranými webovým robotem
   Result: Vyhodnocení shody hodnocení se všemi uživateli pro každého uživatele
 1 Zjištění počtu zpracovávaných uživatelů;
 2 Kontrola, zda již cílová databáze neobsahuje nějaká data;
 3 for Procházej postupně všechny id uživatelů do
 4     Načtení dat o uživateli1 podle id;
 5     for Procházej postupně všechny uživatele s id větším než uživatel1 do
 6         Načtení dat o uživateli2 podle id;
 7         if Pokud má uživatel1 méně záznamů then
 8             foreach Procházej postupně data uživatele1 do
 9                 for Procházej postupně data uživatele2 do
10                     if Data obou uživatelů stejná then
11                         Zpracování dat;
12                     if Pokud mají data uživatele1 větší hodnotu než data uživatele2 then
13                         Skok na řádek 9;
14                     Skok na řádek 8;
15         else
16             foreach Procházej postupně data uživatele2 do
17                 for Procházej postupně data uživatele1 do
18                     if Data obou uživatelů stejná then
19                         Zpracování dat;
20                     if Pokud mají data uživatele2 větší hodnotu než data uživatele1 then
21                         Skok na řádek 14;
22                     Skok na řádek 13;
23         if Pokud byly nalezeny nějaké společné filmy then
24             Konečné zpracování výsledku;
25             Uložení dat do databáze;
26 Ukončení práce programu;
```

Část programu popsaná algoritmem 2 je implementací první metody porovnání zmíněné v kapitole 5.4. Program si nejprve zjistí z tabulky *Uzivatel* počet uživatelů, se kterými bude pracovat.

Dalším krokem 2 se provádí kontrola, zda již nejsou v cílové tabulce *Shoda* některé údaje zapsány a tudíž lze začít práci s daty od posledního uloženého údaje. Tato kontrola se provádí nejprve zjištěním nejvyššího uloženého čísla v atributu *IDA*. Postupně se snižuje id hledaného uživatele, dokud se nenarazí na existující záznam v tabulce. Poté

se tento stejný postup použije i pro atribut *IDB* s vyjímkou, že se hledá nejvyšší hodnota *IDB* pro již zjištěný nejvyšší *IDA*.

Zpracování dat, krok 2, pak začíná s uživatelem1 rovnému nalezené hodnotě *IDA* a uživatel2 má hodnotu o jednu větší než nalezený *IDB*. Pro každého uživatele se provede porovnání se všemi ostatními uživateli, kteří mají vyšší id než daný uživatel.
K porovnávaným uživatelům načítám příslušná data z tabulky *Hodnotil*, kroky 4 a 6.

Krok 5 říká, že se budou procházet jen hráči s vyším id než právě zpracovávaný uživatel, jelikož jeho porovnání s uživateli s menším id je již provedeno z předchozích výpočtů a je již uloženo v databázi.

Při porovnávání se vždy hledají společná data z menšího počtu údajů u porovnávaných dvou uživatelů, krok 7. Pokud jsou porovnávané id filmu u obou stejné, provede se výpočet rozdílu v hodnoceních tohoto filmu danými uživateli. Data pro hledání jsou setříděna vzestupně, proto program může pokračovat v porovnávání tam, kde skončil při minulém porovnávání. Příklad 5.2 ukazuje, jak probíhá porovnání.

Příklad 5.2
Mějme uživatele *U1* s filmy {2, 3, 5} a uživatele *U2* s filmy {1, 2, 4}. Následující kroky popisují všechny možnosti průběhu porovnávání.

1. krok porovnání U1(2) - U2(1)

2. krok porovnání U1(2) - U2(2)

3. krok porovnání U1(3) - U2(4)

4. krok porovnání U1(5) - U2(4) ■

Tato činost je prováděna kroky 8 až 14 a kroky 16 až 22 popisují situaci, kdy má druhý porovnávaný uživatel méně dat, a je proto použit k porovnávání.

Poslední kroky 23 až 25 již jen provedou kontrolu, zda mají porovnávaní dva uživatelé společné filmy. Pokud ano, vypočte se celkové číslo shody dvou uživatelů podělením součtu rozdílů hodnocení počtem společných filmů a výsledek se uloží do tabulky *Shoda*.

Algoritmus 3: Druhá metoda pro porovnávání

```
   Data: Databáze s daty sebranými webovým robotem, tabulka SetrideneFilmy
   Result: Vyhodnocení shody hodnocení se všemi uživateli pro každého uživatele
1  Zjištění počtu zpracovávaných uživatelů;
2  Kontrola, zda již cílová databáze neobsahuje nějaká data;
3  for Procházej postupně všechny id uživatelů do
4      Načtení dat o uživateli1 podle id;
5      foreach Procházej postupně všechny filmy daného uživatele do
6          Načtení všech uživatelů hodnotících daný film;
7          foreach Procházej postupně načtené hodnotící uživatele do
8              Zapisování dat do pomocné tabulky;
9      for Procházej postupně pomocné pole do
10         Konečné zpracování výsledku;
11         Uložení dat do databáze;
12 Ukončení práce programu;
```

Algoritmus 3 realizuje druhou metodu pro porovnávání z kapitoly 5.4. Před začátkem provádění se musí nejprve provést setřídění tabulky *Hodnotil* podle id filmu do tabulky *SetrideneFilmy*. Algoritmus přepsání dat je popsán v algoritmu 4. Jedná se o postupné přepsání dat z jedné do druhé tabulky.

Kroky 1, 2, 3 a 4 jsou podrobně popsány v předešlé kapitole 5.5, jedná se o zjištění počtu zpracovávaných uživatelů a o kontrolu, zda již databáze neobsahuje část zpracovaných dat. Pokud byla již uložena některá data v databázi, pokračuje zpracování od prvního ještě nezpracovaného údaje. Následně načtu data o uživateli, kterého budu porovnávat se všemi ostatními uživateli.

Před krokem 5 je potřeba si vytvořit pomocné pole o velikosti počtu uživatelů. Datovým typem tohoto pole bude struktura *Uzivatel*, sloužící pro ukládání průběhu zpracování. Struktura má dvě proměnné typu integer. První proměnná *pocSpolec* slouží pro uložení počtu společných filmů. Do druhé proměnné *soucetH* se ukládá průběžný stav rozdílu hodnocení jednotlivých společných filmů.

V kroku 6 načítáme údaje, kteří uživatelé hodnotili právě zpracovávaný film, z tabulky *SetrideneFilmy*. Vyhledávají se jen informace o zatím nezpracovaných uživatelích, což jsou uživatelé s vyšším id než je id právě zpracovávaného uživatele. Podobnost uživatelů s nižším id byla zpracována při dřívějším porovnávání těchto uživatelů.
Po načtení těchto dat, jsou tato data postupně zpracovávána. Získaná data jsou zapsána do pomocné tabulky. Tyto operace provádí kroky 7 a 8.

Kroky 9, 10 a 11 provádějí závěrečné zpracování před začátkem zpracování dalšího uživatele. Postupně se prochází pomocné pole, kde každý záznam odpovídá jednomu uživateli. Pokud mají právě zpracovávaní dva uživatelé společné některé filmy, provede

se finální výpočet shody hodnocení těchto dvou uživatelů a data jsou zapsána do tabulky *Shoda*.

Algoritmus 4: Přetřídění dat podle IDF

Data: Tabulka *Hodnotil*
Result: Tabulka *SetrideneFilmy*, v které jsou data z tabulky *Hodnotil* uspořádány podle id filmu

```
1 Zjištění počtu zpracovávaných uživatelů;
2 for Procházej postupně všechna id uživatelů do
3     Načtení dat, souvisejících s uživatelem, z tabulky Hodnotil;
4     foreach Procházej postupně všechna načtená data do
5         Ukládání dat do nové tabulky SetrideneFilmy;
```

5.6 Analýza programu pro prezentaci výsledků

Tento program má sloužit k zobrazení výsledků práce předchozích dvou programů. Aby ukázal práci webového robota, bude umožňovat výpis informací o vybraném uživateli.

Dále bude prezentovat samotný výpočet podobnosti mezi zadanými dvěma uživateli. Pro výpočet podobnosti bude využit první algoritmus popsaný dříve v Analýze programu pro zpracování dat. Díky poslední finkci programu tak bude možné porovnat, zda oba algoritmy získávají stejné výsledky.

Poslední funkcionalitou bude totiž výpis podobnosti z databáze získané programem pro zpracování dat. Naplnění databáze probíhalo pomocí druhého algoritmu. Uživatel si bude moct vybrat, o kterého uživatele se zajímá, a nastavit parametry výsledku. Může nastavit maximální počet vypsaných uživatelů a také počet společných hodnocených filmů uživatelů.

5.6.1 Databáze programu pro prezentaci výsledků

Tento program bude také využívát databáze na MS SQL Serveru 2008. První databáze bude databáze naplněná webovým robotem a druhá bude obsahovat tabulku všech podobností vypočtenou programem pro zpracování dat.

Z databáze naplněné webovým robotem bude využívat tabulku Uzivatel, když bude vypisovat všechny informace o uživateli v této tabulce. Pro výpočet podobnosti mezi dvěma uživateli pak bude potřebovat tabulky Uživatel a Hodnotil. Výpočet bude potřebovat hodnoty atributů IDU, IDF a Hodnoceni.

Databáze naplněná programem pro zpracování dat pak bude využita pro výpis uživatelů s podobným vkusem. Program bude potřebovat atributy IDA, IDB, Shoda a PocSpol z tabulky Shoda.

Jednotlivé tabulky a atributy tabulek jsou podrobněji popsány v předchozích kapitolách.

5.6.1.1 Funkční analýza V tomto programu jsou tři elementární funkce. Jedna pro výpis informací o uživateli. Druhá pro přímý výpočet podobnosti mezi dvěma uživateli. Poslední slouží pro zobrazení uživatelů s nejpodobnějším hodnocením filmů k danému vybranému uživateli.

Minispecifikace výpisu uživatelů s nejpodobnějším hodnocením:

1. Uživatel zadá přezdívku nebo id uživatele
2. Uživatel zadá maximální počet zobrazených záznamů
3. Uživatel zadá minimální počet společných hodnocených filmů
4. Uživatel klikne na tlačítko Výpis podobnosti
5. Načti hodnotu přezdívky nebo id
6. Zjisti, zda jde o id nebo přezdívku
   (a) Byla-li zadána přezdívka, zjisti zda neobsahuje nepovolené znaky
   (b) Pokud přezdívka obsahuje nepovolené znaky, vypiš chybovou hlášku a skonči
7. Ověř id nebo přezdívku v databázi
8. Nebyl-li odpovídající záznam nalezen, vypiš chybovou hlášku a skonči
9. Načti hodnotu maximálního počtu uživatelů
10. Není-li hodnota maximálního počtu uživatelů číslo od 1 do 100, vypiš chybou hlášku a skonči
11. Načti minimální počet filmů
12. Není-li hodnota minimálního počtu filmů kladné celé číslo, vypiš chybou hlášku a skonči
13. Proveď dotaz do databáze se zadanými parametry
14. Vypiš data ze serveru

6 Zhodnocení výsledků

V této kapitole uvedu výsledky měření délky provádění některých klíčových operací. Testy byly prováděny na školním serveru epe.vsb.cz, na počítači byl nainstalován systém Microsoft Windows Server 2003 s procesorem Intel Xeon 2,66GHz a RAM 8GB. Pro dokončení práce, musel být projekt přemístěn na školní server argexpr.vsb.cz. Důvody přesunu jsou zmíněny později. Nový server pak má systém Windows Server Datacenter, procesor AMD Opteron 1,80GHz a paměť RAM 31,9GB. Pro měření času bylo použito třídy Stopwatch a jejich funkcí.

6.1 Vyhodnocení práce webového robota

Doba trvání načtení stránky je závislá na velikosti obsahu stránky. Běžná stránka s výpisem informací o velikosti 42kB se načítá 266ms. Stránky s výpisy seznamů mají velikost kolem 270kB a načítají se 2024ms. V tabulce 15 jsou vypsány informace o průběhu stahování, kde je uveden počet uživatelů, jejiž hodnocení bylo ukládáno, počet hodnocení, která se musela projít a nakonec délka trvání zpracování těchto dat. Z grafu 2, který je vytvořen nad daty v tabulce 15, je pak patrné zrychlení stahování v závislosti na velikosti již zpracovaných dat. To je způsobeno tím, že když už je uloženo mnoho filmů v databázi tak nedochází k načítání jejich stránek a následnému zpracování.

Celková doba stahování všech dat pak není známa, jelikož došlo během stahování dat k dvěma restartům počítače, v důsledku nahrání aktualizací na daný počítač. Program se pak po opětovném spuštění dostane na místo, kde skončil, rychleji, avšak musí stejně projít hodně dat znovu.

6.2 Vyhodnocení práce programu pro zpracování dat

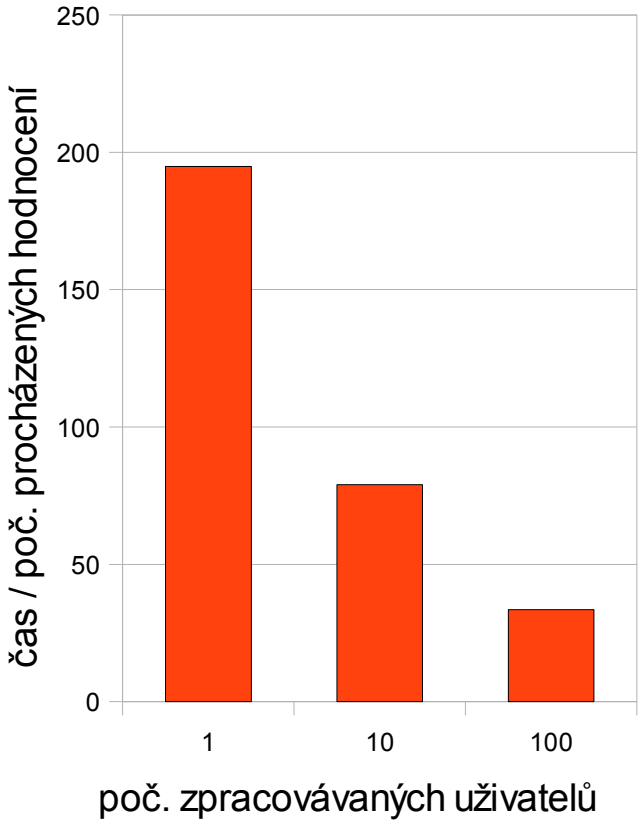
Tabulka 16 zobrazuje, jak dlouho trvá oběma metodám zpracovat podobnost daných uživatelů. Z tabulky je patrné, že první metoda je značně pomalá, a to hlavně kvůli svému častému dotazování do databáze. Proto také byla navrhnuta druhá metoda, která využívá stejnou podobnostní funkci, avšak liší se ve způsobu zpracování dat, viz kapitola 5.4.

Obě metody pracují se získanými hodnoceními filmu, kterých je 16777702. A z těchto dat musí vypočítat podobnost mezi všemi uživateli. Rychlost výpočtu podobnosti obou metod je patrná v grafu 3, kde je uvedeno, jak dlouho trvalo oběma metodám porovnat stejné množství uživatelů.

Protože první metoda byla velmi pomalá, byl program ukončen dříve než došlo ke zpracování všech dat. Pomocí druhé metody již bylo dosaženo požadovaného porovnání a výpočtu podobnosti vkusu mezi všemi uživateli. Průběh celkového zpracování pomocí druhé metody je zobrazen v grafu 4. Graf udává, kolik uživatelů již mělo vypočítány všechny podobnosti v zadaný den. V tabulce 17 jsou vypsány počty uživatelů s hotovým porovnáním v uvedené dny. Číslo již zpracovaných uživatelů roste s časem rychleji v důsledku menšího počtu porovnání u každého dalšího uživatele a také je to dáno tím, že později zpracovávaní uživatelé mají méně hodnocení filmů.

uživatel	poč. hodnocení	délka trvání (ms)
1	4924	959633
10	40130	3171796
100	329586	11020715

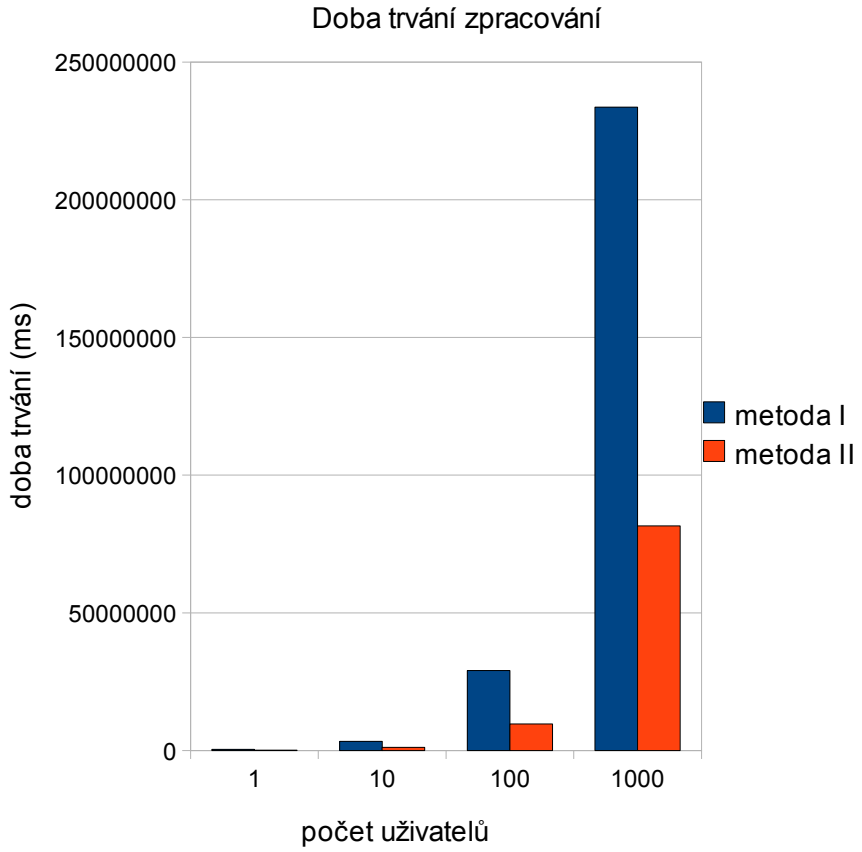
Tabulka 15: Tabulka trvání stahování dat z webu u 1, 10 a 100 uživatelů



Obrázek 2: Graf závislost rychlosti stahování na velikosti již zpracovaných dat

	1 uživatel	10 uživatelů	100 uživatel	1000 uživatelů
I metoda	381839 ms	3282828 ms	29011487 ms	233567772 ms
II metoda	162011 ms	1102344 ms	9631537 ms	81531126 ms

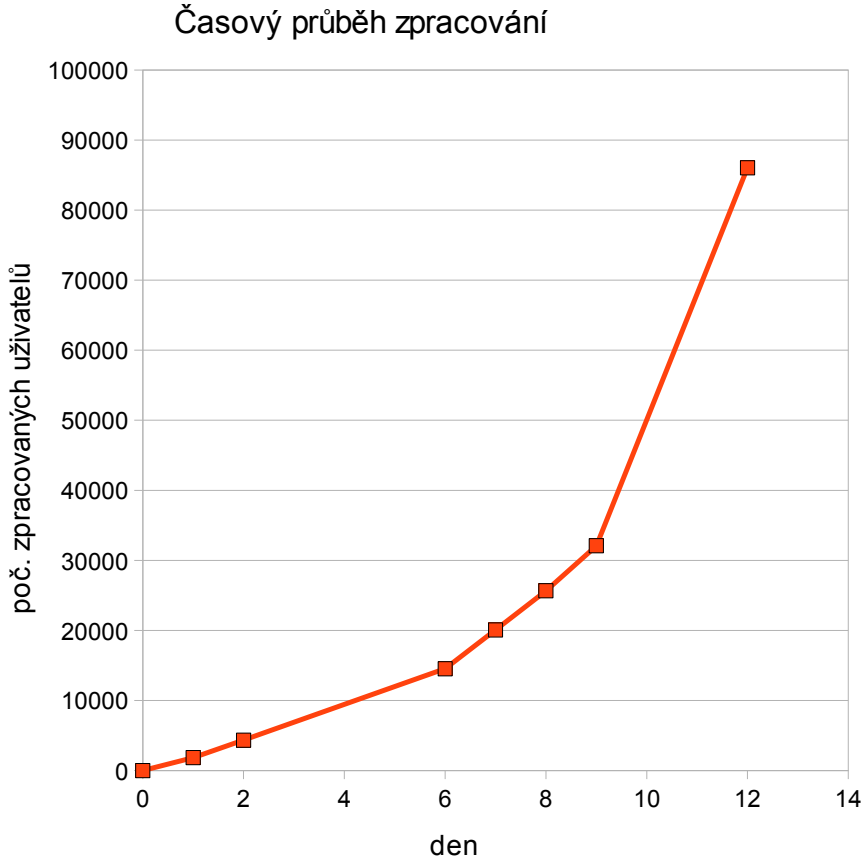
Tabulka 16: Tabulka trvání vyhodnocení dat u daného počtu uživatelů obou metod



Obrázek 3: Časový graf průběhu zpracování u jednotlivých metod

den	1	2	6	7	8	9
poč. uživatelů	1853	4334	14532	20069	25668	32086

Tabulka 17: Tabulka porovnaných uživatelů v závisloti na čase



Obrázek 4: Časový graf průběhu výpočtu podobnosti u uživatelů

6.2.1 Zhodnocení výsledku programu pro zpracování dat

Jak již bylo zmíněno v kapitole Databáze programu pro zpracování dat, výsledkem bude velké množství vypočtených podobností mezi jednotlivými uživateli. Pro zvolený způsob uložení by mělo být v databázi pro 86029 uživatelů přes $3,7$ miliardy podobností. Do databáze však nejsou ukládány podobnosti mezi uživateli bez společného filmu. Výsledný počet záznamů v tabulce je pak 1970145458. Tato tabulka, do které se ukládají tři proměnné typu integer a jedna typu float, zabírá 55GB.

Problém pak nastává při dotazování do databáze na výpis podobností k danému uživateli. Dotaz je příliš složitý a vyprší čas čekaní na odpověď. Problém je ve způsobu uložení dat, kde kvůli úspory místa je každá podobnost pro dané dva uživatele uložena jen jednou. Pro rychlé vyhledání by musela být daná podobnost uložena ještě jednou, ale s prohozeným prvním a druhým uživatelem. Uložení dat je patrné v tabulce 18.
Tabulka slouží pouze pro grafickou ukázku problému s uložením dat. Žádný uživatel není samozřejmě porovnáván sám se sebou. U prvního uživatele problém při vyhledávání podobnosti není. První uživatel je u všech podobností jako Uživatel1. Druhý uživatel však již nemá přímo přístupnou informaci o podobnosti mezi ním a prvním uživatelem. Poslední uživatel pak musí všechny své podobnosti hledat u předchozích uživatelů. Právě tato část dotazu způsobí dlouhé vyhledávání v tabulce, jelikož je tabulka indexována podle prvního uživatele. Vyhledávání z pozice druhého uživatele musí pak projít mnoho záznamů.

Tento problém může být řešen dvěma způsoby. První je doplnění tabulky tak,
aby každý uživatel měl přístupné všechny své podobnosti z místa Uživatele1. Tento způsob však zdvojnásobí velikost stávající tabulky. Velikost tabulky by tak byla 110GB. Nebo nezatěžovat databázi jedním složitým dotazem, ale pomocí více jednoduchých dotazů a zpracování získaných dat dosáhnout požadovaných údajů. Toto řešení však bude časově mnohem náročnější. Lepší je tedy první varianta, jelikož dlouhé čekání na vypsání výsledků je nežadoucí.

Nejprve byla vyzkoušena metoda více dotazů, avšak ukázalo se, že je vážně hodně pomalá. Za den zvládla setřídit data přibližně pro 4000 uživatelů, další uživatele by se zpracovávali stejně rychle. Proto byl tento algoritmus zastaven.

Místo něho bylo provedeno dopsání chybějících záznamů pro zkompletování výsledné tabulky, aby bylo možné získat data jediným dotazem. Dopsání je provedeno jednoduše postupným načítáním jednotlivých údajů z tabulky a přehozením id uživatelů mezi sebou. Hodnota podobnosti a počtu filmů zůstává stejná. I u této metody, však bylo naraženo na výše zmiňovaný problém a to velikost databáze. Bylo spotřebováno veškeré volné místo na serveru, před dokončením doplnění tabulky. Proto byl projekt i s databází přesunut na server *argexpr.vsb.cz*.

Avšak tak velká databáze není vhodná pro prezentaci výsledků. Byla proto vytvořená nová databáze, do které se uložilo ke každému uživateli jen sto uživatelů s nejpodobnějším ohodnocením společných filmů. Z nové databáze již není problém potřebná data získat pomocí jediného dotazu. Tato nová menší databáze je pak využita v programu pro prezentaci výsledků. Databáze tak omezuje možnost vyběru počtu uživatelů při výpisu nejpodobnějších uživatelů.

Uživatel1 \ Uživatel2	**1. uživatel**	**2. uživatelů**	**3. uživatel**	**4. uživatelů**
1. uživatel	X	0,2	1,35	0
2. uživatel		X	0,62	4,7
3. uživatel			X	0,01
4. uživatel				X

Tabulka 18: Tabulka zobrazující způsob uložení dat podobnosti v databázi

7 Závěr

Následující text shrnuje získané poznatky a výsledky během vypracovávání této práce. Vytvořené aplikace pro sběr dat z filmové sociální sítě, výpočet podobnosti mezi uživateli a zobrazení výsledků podobnosti mezi uživateli plní své stanovené funkce. Program pro získání dat z filmové sociální sítě (webový robot) vyhledává požadovaná data na stránkách ČSFD a ukládá je do databáze. Data této databáze jsou pak vstupními daty pro program pro výpočet podobnosti mezi uživateli. Program určuje podobnost mezi uživateli na základě společných filmů mezi uživateli a rozdílu ohodnocení těchto filmů. Výsledky jsou pak opět uloženy do databáze. Poslední program pak slouží pro výpis podobností mezi uživateli.

Hlavními problémy byla rychlost provádění stahování dat ze stránek a výpočtů podobností. U stahování je to způsobeno množstvím stránek, které je třeba projít. U výpočtů podobností jsou hlavním omezujícím prvkem dotazy pro manipulaci s daty z databáze. Dalším problémem byla velikost výsledné databáze porovnání. Nakonec byla výsledná databáze transformována na pouze sto uživatelů s nejpodobnějším hodnocením filmů ke každému uživateli.

Pro mne je osobním přínosem osvojení si práce s dotazovacím jazykem LINQ a obeznámení se s problematikou podobnostního vyhledávání v datech a s tím souvisejících metrických a nemetrických funkcí pro výpočet dané podobnosti.

Možná vylepšení bych viděl v urychlení jednotlivých problemových částí programů. Také by bylo zajímavé vymyslet a naprogramovat jiná porovnání (podobnosti) mezi uživateli a porovnat výsledky.

Bc. Pavel Lednický

8 Reference

[1] DONÁT, Jiří, *Sociální sítě – cesta ke strukturovanějšímu Internetu?*, [online], 2006-02-22, [cit. 2010-04-09], Dostupné na WWW <http://www.lupa.cz/clanky/socialni-site-cesta-ke-strukturovanejsimu-internetu/>.

[2] Wikipedia, *List of social networking websites*, [online], [cit. 2010-04-09], Dostupné na WWW <http://en.wikipedia.org/wiki/List_of_social_networking_websites>.

[3] The Internet Movie Database, , [online], [cit. 2010-04-09], Dostupné na WWW <http://www.imdb.com/>.

[4] Česko-Slovenská filmová databáze, , [online], [cit. 2010-04-09], Dostupné na WWW <http://www.csfd.cz/>.

[5] Filmová databáze, , [online], [cit. 2010-04-09], Dostupné na WWW <http://www.fdb.cz/>.

[6] Wikipedia, *Internetový robot*, [online], [cit. 2010-04-09], Dostupné na WWW <http://cs.wikipedia.org/wiki/Internetový_robot>.

[7] KORANDA, Petr, *SEO - jak pracují vyhledávací roboti*, [online], 2008-04-14, [cit. 2010-04-09], Dostupné na WWW <http://www.peakpointnet.cz/cz/piseme/clanky/seo-jak-pracuji-vyhledavaci-roboti>.

[8] KORANDA, Petr, *SEO - optimalizace pro vyhledávače*, [online], 2008-03-27, [cit. 2010-04-09], Dostupné na WWW <http://www.peakpointnet.cz/cz/piseme/clanky/seo-optimalizace-pro-vyhledavace>.

[9] BYDŽOVSKÁ, Hana, *Podobnost digitálních obrázků pomocí Earth Mover's Distance*, Bakalářská práce na fakultě Informatiky Masarykovy Univerzity, 2007, Vedoucí bakalářské práce RNDr. David Novák, Dostupné na WWW <http://is.muni.cz/th/139544/fi_b/bc.pdf>.

[10] SKOPAL, Tomáš, BUSTOS, Benjamin, *On Nonmetric Similarity Search Problems in Complex Domains*, Clanek: objeví se v ACM Computing Surveys, Vydavatel: ACM, 2010.

[11] SKOPAL, Tomáš, *Vyhledávání v multimediálních databázích: Modelování a podobnost*, [online], Přednášky k předmětu Vyhledávání v multimediálních databázích na fakultě Matematiky a Fyziky Karlovy Univerzity, Dostupné na WWW <http://siret.ms.mff.cuni.cz/skopal/DBI030.htm>.

[12] HANÁK,Ján, *C# 3.0: Programovaní na platformě .NET 3.5.*, Brno: Zoner Press, 2009, 288 s, [cit. 2010-04-09], ISBN 978-80-7413-046-5.

[13] Microsoft, *LINQ*, [online], [cit. 2010-04-09], Dostupné na WWW <http://msdn.microsoft.com/en-us/netframework/aa904594.aspx>.

[14] Wikipedia, *LINQ*, [online], [cit. 2010-04-09], Dostupné na WWW <http://cs.wikipedia.org/wiki/LINQ>.

[15] ŠARMANOVÁ, Jana, *Teorie zpracování dat*, [online], Ostrava: VŠB- Technická univerzita Ostrava, 2007 [cit. 2010-04-19]. Dostupné z WWW <http://www.elearn.vsb.cz/>. ISBN 978-80-248-1498-8.

A Obsah přiloženého DVD-ROM

- Text práce
- Vytvořené programy (webový robot, výpočet podobnosti, vizualizace podobnosti
- Programátorské a uživatelské příručky k programům
- Databáze pro práci programů